Active Area Coverage from Equilibrium

Ian Abraham, Ahalya Prabhakar, and Todd D. Murphey [*]

Department of Mechanical Engineering, Northwestern University,
2145 Sheridan Road, Evanston, IL 60208, USA
i-abr@u.northwestern.edu, a-prabhakar@u.northwestern.edu,
t-murphey@northwestern.edu

Abstract. This paper develops a method for robots to integrate stability into actively seeking out informative measurements through coverage. We derive a controller using hybrid systems theory that allows us to consider safe equilibrium policies during active data collection. We show that our method is able to maintain Lyapunov attractiveness while still actively seeking out data. Using incremental sparse Gaussian processes, we define distributions which allow a robot to actively seek out informative measurements. We illustrate our methods for shape estimation using a cart double pendulum, dynamic model learning of a hovering quadrotor, and generating galloping gaits starting from stationary equilibrium by learning a dynamics model for the half-cheetah system from the Roboschool environment.

Keywords: active exploration, safe learning, active learning

1 Introduction

Robot learning has proven to be a challenge in real-world application. This is partially due to the ineffectiveness of passive data acquisition for learning and a necessity for action in order to generate informative data. What makes this problem even more challenging is that active data gathering is not a stable process. It involves exciting states in order to acquire new information. Safe exploration then becomes a challenge for modern day robotics. The problem becomes exacerbated when memory and task constraints (i.e., actively collecting data after deployment) are imposed on the robot. If the structures that compose the dynamics of the robot change over time, the robot will need to explore its own dynamics in a manner that is systematic and informative, avoiding damage to the underlying structures (and humans) in the environment. In this paper, we address these fundamental issues by developing an algorithm that is inspired by hybrid systems theory. This algorithm enables robots to actively pursue informative data by generating area coverage while guaranteeing Lyapunov attractiveness during exploration.

Active data acquisition and learning are often considered part of the same problem of learning from experience [1, 2]. This is generally seen in the field of reinforcement learning (RL) where attempts at a task, as well as learning from

[*] This material is based upon work supported by the National Science Foundation under Grants CNS 1837515. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the aforementioned institutions.

the outcome of actions, are used to both learn policies and predictive models [1, 3]. As a result, generalizing these methods to real-world application has been a topic of research [1, 3, 4] where data-inefficiency dominates much of the progress. A solution to the problem of data-inefficiency is to simulate robots in a realistic virtual environment and subsequently use the large amount of synthetic data to solve a learning problem before applying the results on a real robot [5]. This leads to issues such as the "reality-gap" where finer modelling details such as motor delays lead to poor quality data for learning.

Existing work addresses the data-inefficiency problem by actively seeking out informative data using information maximization [6] or by pruning a data-set based on some information measure [7]. These methods still suffer from the problem of local minima due to a lack of exploration or non-convex information objectives [8]. Safety in the task is also a concern when actively seeking out informative measurements. Methods typically provide some bound on the worst outcome model using probabilistic approaches [9], but often only consider the safety with respect to the task and not with respect to the data collection process. We focus on problems where data collection involves exploring the state-space of robots where safe generation of informative data is important. In treating data acquisition as a dynamic area coverage problem—where the time spent during the trajectory of the robot is proportional to regions where there is an expectation of informative data—we are able to uncover more informative data that is not already expected. With this approach, we can provide attractiveness guarantees—that the robot will eventually return to a stable state—while providing control authority that allows the robot to actively seek out informative data in order to later solve a learning task. *Thus, our contribution is an approach to dynamic area coverage for active data collection that starts from equilibrium policies for robots.*

We structure the paper as follows: Section 2 provides a list of related work, Section 3 defines the problem statement for this work. Section 4 formulates the algorithm for active data acquisition from equilibrium. Section 5 provides simulated and experimental examples of our method. Last, Section 6 provides concluding remarks on our method and future directions.

2 Related Work

Existing work generally formulates problems of active data acquisition as information maximizing with respect to a known parameterized model [10, 11]. The problem with this approach is that robots need to address local optima [11, 12], resulting in insufficient data collection. Other approaches have sought to solve this problem by thinking of information maximization as an area coverage problem [12, 13]. Ergodic exploration, in particular, has remedied the issue of local extrema by using the ergodic metric to minimize the Sobelov distance [14] from the time-averaged statistics of the robot's trajectory to the expected information in the explored region. This enables both exploration (quickly in low information regions) and exploitation (spending significant amount of time in highly informative regions) in order to avoid local extrema and collect informative measurements. The major downside is that this method assumes that the model of the robot is fully known. Moreover, there is little guarantee that the robot will not destabilize during the exploration process. This becomes an issue when the robot must explore part of its own state-space (i.e., velocity space) in order to generate informative data. To the authors' best knowledge this has not been done to this date. Another issue is that these methods do not scale well with the dimensionality of the search space, making experimental applications with this approach challenging due to computational limitations.

Our approach overcomes these issues by using a sample-based KL-divergence measure [13] as a replacement for the ergodic metric. This form of measure has been used previously; however, it relied on motion primitives in order to compute control actions [13]. We avoid this issue by using hybrid systems theory in order to compute a controller that sufficiently reduces the KL-divergence measure from an equilibrium stable policy. As a result, we can use approximate models of dynamical systems instead of complete dynamic reconstructions in order to actively collect data while ensuring safety in the exploration process through a notion of attractiveness.

The following section formulates the problem statement that our method solves.

3 Problem Statement

Modeling Assumptions and Stable Policies Assume we have a robot whose approximate dynamics can be modeled using

$$\dot{x} = f(x, u) = g(x) + h(x)u \tag{1}$$

where $x \in \mathbb{R}^n$ is the state of the robot, $u \in \mathbb{R}^m$ is a control vector applied to the robot, $g(x) : \mathbb{R}^n \to \mathbb{R}^n$ is the free unactuated dynamics, $h(x) : \mathbb{R}^n \to \mathbb{R}^{n \times m}$ is the actuated dynamics, and $\dot{x}$ is the time rate of change of the robot at state x subject to the control u. Moreover, let us assume that there exists a Lyapunov function $V(x)$ such that under a policy $\mu(x) : \mathbb{R}^n \to \mathbb{R}^m$, $\dot{V}(x) < 0 \ \forall x \in \mathcal{B}$, where $\mathcal{B} = \{x \in \mathbb{R}^n | \|x\| < r\}$ for $r > 0$. For the rest of the paper, we will refer to $\mu(x)$ as an equilibrium policy.

KL-divergence and Area Coverage Given the assumptions of known approximate dynamics and the equilibrium policy, we can define active exploration for informative data acquisition as automating safe switching between $\mu(x)$ and some control authority $\mu_\star(t)$ that generates actions that actively seek out informative data. This is accomplished by specifying the active data acquisition task using an area coverage objective where we minimize the KL-divergence between the time average statistics of the robot along a trajectory and a spatial distribution defining the current coverage requirement. We can then define an approximation to the spatial statistics of the robot as follows:

Definition 1. *Given a search domain $\mathcal{X}^v \subset \mathbb{R}^{n+m}$ where $v \leq n + m$, the Σ-approximated time-averaged statistics of the robot, i.e., the time the robot spends in regions of the search domain $\mathcal{X}^v$, is defined by*

$$q(s \mid x(t), \mu(t)) = \frac{\eta}{T_r} \int_{t_i - t_r}^{t_i + T} \exp\left[-\frac{1}{2} \left(s - x_v(t) \right)^\top \Sigma^{-1} \left(s - x_v(t) \right) \right] dt \tag{2}$$

where $s \in \mathcal{X}^v \subset \mathbb{R}^{n+m}$ is a point in the search domain $\mathcal{X}^v$, $x_v(t)$ is the component of the robot's trajectory $x(t)$ and actions $\mu(t)$ that intersects the search domain $\mathcal{X}^v$, $\Sigma \in \mathbb{R}^{v \times v}$ is a positive definite matrix parameter that specifies the width of the Gaussian, η is a normalization constant such that $q(s) > 0$ and $\int_{\mathcal{X}^v} q(s)ds = 1$, t_i is the i^{th} sampling time, and $T_r = T + t_r$ is sum of the time horizon T and amount of time t_r the robot remembers $x_v(t)$ into the past.

This is an approximation because the true time-averaged statistics, as described in [12], is a collection of delta functions parameterized by time. We approximate the delta function as a Gaussian distribution with covariance Σ, converging as $\|\Sigma\| \to 0$. Using this approximation, we are able to relax the ergodic area-coverage objective in [12] and use the following KL-divergence objective [13]:

$$D_{\text{KL}}(p\|q) = \int_{\mathcal{X}^v} p(s) \ln \frac{p(s)}{q(s)} ds = \mathbb{E}_{p(s)} \left[\ln p(s) - \ln q(s) \right],$$

where $\mathbb{E}$ is the expectation operator, $q(s) = q(s \mid x(t), \mu(t))$, and $p(s)$, $p(s) > 0, \int_{\mathcal{X}^v} p(s)ds = 1$, is a distribution that describes where in the search domain an informative measurement is likely to be acquired. We can further approximate the KL-divergence via sampling where we approximate the expectation operator as

$$D_{\text{KL}}(p\|q) = \mathbb{E}_{p(s)} \left[\ln p(s) - \ln q(s) \right] \approx \sum_{i=1}^{N} p(s_i) \ln p(s_i) - p(s_i) \ln q(s_i), \tag{3}$$

where N is the number of samples in the search domain drawn from a uniform distribution. With this formulation, we can approximate the ergodic coverage metric using (3).

In addition to the KL-divergence, we can add a task objective

$$J_{\text{task}} = \int_{t_i}^{t_i+T} \ell(x(t), \mu(x(t)))dt + m(x(t_i + T)) \tag{4}$$

where t_i is the i^{th} sampling time, T is the time horizon, $\ell(x, u) : \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ is the running cost, $m(x) : \mathbb{R}^n \to \mathbb{R}$ is the terminal cost, and $x(t) : \mathbb{R} \to \mathbb{R}^n$ is the state of the robot at time t. This additional objective will typically encode some other task, in addition to the KL-divergence objective.

By summing the KL-divergence objective and a task objective (4), we can then pose active data acquisition as an optimal control problem subject to the initial approximate dynamic model of the robot. More formally, the objective is written as

$$J = D_{\text{KL}}(p\|q) + \int_{t_i}^{t_i+T} \ell(x(t), \mu(x(t)))dt + m(x(t_i + T)) \tag{5}$$

where the goal is to generate a control $\mu_\star(t)$ that minimizes (5) subject to the approximate dynamics (1). Because we are including the equilibrium policy in the objective, we are able to synthesize controllers that take into account the equilibrium policy $\mu(x)$ and the desire to actively seek out measurements.

For the rest of the paper, we assume the following:

- We have an initial approximate model $\dot{x} = f(x, u)$ of the robot.
- We also have an initial policy $\mu(x)$ that maintains the robot at equilibrium, for which there is a Lyapunov function.

These two assumptions are reasonable in that often robots are designed around stable states and typically have locally stable policies.

The following section uses the fact that we have an initial policy $\mu(x)$ in order to synthesize control vectors $\mu_\star(t) : \mathbb{R} \to \mathbb{R}^m$ that reduce (5). Specifically, we want to generate a hybrid composition of control actions that enable active data collection and actions that stabilize the robotic system. That way, it is possible to quantify how much the robot is deviating from a stable equilibrium. Thus, we motivate using hybrid systems theory in order to consider how much the objective (5) changes from switching from the equilibrium policy $\mu(x(t))$ to the control $\mu_\star(t)$. By quantifying the change, we specify an unconstrained optimization which solves for a control $\mu_\star$ that applies actions that retain Lyapunov attractiveness.

4 Algorithm

Our algorithm starts by considering the objective defined in (5) subject to the approximate dynamic constraints (1) and policy $\mu(x)$. We want to quantify how sensitive the objective is to switching from policy $\mu(x(t))$ to the control vector $\mu_\star(t)$ for time $\tau \in [t_i, t_i + T]$ for a infinitesimally small time duration λ. This sensitivity will be a function of $\mu_\star$ and inform us of the most influential time to apply $\mu_\star(t)$. Thus, we can use the sensitivity to write an objective whose minimizer is the schedule of control vectors $\mu_\star(t)$ that reduces the objective (5).

Proposition 1. *The sensitivity of the objective (5) with respect to the duration time λ, of switching from the policy $\mu(x)$ to the control $\mu_\star(t)$ at time τ is*

$$\frac{\partial J}{\partial \lambda}\bigg|_{t=\tau} = \rho(\tau)^\top (f_2 - f_1) \tag{6}$$

where $f_2 = f(x(t), \mu_\star(t))$ and $f_1 = f(x(t), \mu(x(t))$, and $\rho(t) \in \mathbb{R}^n$ is the adjoint, or co-state variable which is the solution of the following differential equation

$$\dot{\rho} = -\left(\frac{\partial \ell}{\partial x} + \frac{\partial \mu}{\partial x}^\top \frac{\partial \ell}{\partial u} - \frac{\eta}{T_r} \sum_i \frac{p(s_i)}{q(s_i)} \left(\frac{\partial g}{\partial x} + \frac{\partial \mu}{\partial x}^\top \frac{\partial g}{\partial u} \right) \right) - \left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial u} \frac{\partial \mu}{\partial x} \right)^\top \rho \tag{7}$$

subject to the terminal constraint $\rho(t_i + T) = \frac{\partial}{\partial x} m(x(t_i + T))$.

Proof. Taking the derivative of the objective (5) with respect to the duration time λ gives

$$\frac{\partial}{\partial \lambda} J = \frac{\partial}{\partial \lambda} D_{\text{KL}} + \frac{\partial}{\partial \lambda} J_{\text{task}}.$$

The term $\frac{\partial}{\partial \lambda} D_{\mathrm{KL}}$ is calculated by

$$\begin{aligned}\frac{\partial}{\partial \lambda} D_{\mathrm{KL}}\bigg|_{t=\tau} &= -\sum_i \frac{p(s_i)}{q(s_i)} \frac{\eta}{T_r} \int_{\tau+\lambda}^{t_i+T} \left(\frac{\partial g}{\partial x} + \frac{\partial \mu}{\partial x}^\top \frac{\partial g}{\partial u} \right)^\top \frac{\partial x}{\partial \lambda} dt \\ &= -\sum_i \frac{p(s_i)}{q(s_i)} \frac{\eta}{T_r} \int_{\tau+\lambda}^{t_i+T} \left(\frac{\partial g}{\partial x} + \frac{\partial \mu}{\partial x}^\top \frac{\partial g}{\partial u} \right)^\top \Phi(t,\tau) dt \, (f_2 - f_1) \qquad (8)\end{aligned}$$

where $g = g(s_i \mid x(t), \mu(x(t))) = \exp\left[-\frac{1}{2}\left(s_i - x_s(t)\right)\Sigma^{-1}\left(s_i - x_s(t)\right)\right]$, and $\Phi(t,\tau)$ is the state transition matrix for the integral equation

$$\frac{\partial x}{\partial \lambda} = (f_2 - f_1) + \int_{\tau+\lambda}^{t_i+T} \left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial u} \frac{\partial \mu}{\partial x} \right)^\top \frac{\partial x}{\partial \lambda} dt \qquad (9)$$

where $f_2 = f(x(\tau), \mu_\star(\tau))$ and $f_1 = f(x(\tau), \mu(x(\tau))$.

We can similarly show that the term $\frac{\partial}{\partial \lambda} J_{\mathrm{task}}$ is given by

$$\begin{aligned}\frac{\partial}{\partial \lambda} J_{\mathrm{task}}\bigg|_{t=\tau} &= \int_{\tau+\lambda}^{t_i+T} \left(\frac{\partial \ell}{\partial x} + \frac{\partial \mu}{\partial x}^\top \frac{\partial \ell}{\partial u} \right)^\top \frac{\partial x}{\partial \lambda} dt. \\ &= \int_{\tau+\lambda}^{t_i+T} \left(\frac{\partial \ell}{\partial x} + \frac{\partial \mu}{\partial x}^\top \frac{\partial \ell}{\partial u} \right)^\top \Phi(t,\tau) dt \, (f_2 - f_1) \qquad (10)\end{aligned}$$

using the same expression in (9). Combining (8) and (10) and taking the limit as $\lambda \to 0$ gives

$$\frac{\partial}{\partial \lambda} J = \int_{\tau}^{t_i+T} \left(\frac{\partial \ell}{\partial x} + \frac{\partial \mu}{\partial x}^\top \frac{\partial \ell}{\partial u} - \frac{\eta}{T_r} \sum_i \frac{p(s_i)}{q(s_i)} \left(\frac{\partial g}{\partial x} + \frac{\partial \mu}{\partial x}^\top \frac{\partial g}{\partial u} \right) \right)^\top \Phi(t,\tau) dt \, (f_2 - f_1) \, . \qquad (11)$$

Setting

$$\rho(\tau)^\top = \int_{\tau}^{t_i+T} \left(\frac{\partial \ell}{\partial x} + \frac{\partial \mu}{\partial x}^\top \frac{\partial \ell}{\partial u} - \frac{\eta}{T_r} \sum_i \frac{p(s_i)}{q(s_i)} \left(\frac{\partial g}{\partial x} + \frac{\partial \mu}{\partial x}^\top \frac{\partial g}{\partial u} \right) \right)^\top \Phi(t,\tau) dt$$

and from [15] we can show that (11) can be written as

$$\frac{\partial}{\partial \lambda} J \bigg|_{t=\tau} = \rho(\tau)^\top (f_2 - f_1)$$

where

$$\dot{\rho} = -\left(\frac{\partial \ell}{\partial x} + \frac{\partial \mu}{\partial x}^\top \frac{\partial \ell}{\partial u} - \frac{\eta}{T_r} \sum_i \frac{p(s_i)}{q(s_i)} \left(\frac{\partial g}{\partial x} + \frac{\partial \mu}{\partial x}^\top \frac{\partial g}{\partial u} \right) \right) - \left(\frac{\partial f}{\partial x} + \frac{\partial f}{\partial u} \frac{\partial \mu}{\partial x} \right)^\top \rho.$$

subject to the terminal condition $\rho(t_i + T) = \frac{\partial}{\partial x} m(x(t_i + T))$. □

The sensitivity $\frac{\partial}{\partial \lambda} J$ is known as the mode insertion gradient [15]. We can directly compute the mode insertion gradient for any control $\mu_\star$ that we choose. However, our goal is to find one such control $\mu_\star$ that reduces the objective (5) but still maintains its value near the equilibrium policy $\mu(x)$. To solve for this control, we formulate the following objective function

$$J_2 = \int_{t_i}^{t_i+T} \frac{\partial}{\partial \lambda} J \Big|_{t=\tau} + \frac{1}{2} \| \mu_\star(t) - \mu(x(t)) \|_R^2 \tag{12}$$

where $R \in \mathbb{R}^{m \times m}$ is a positive definite matrix that penalizes the deviation from the policy $\mu(x)$.

Proposition 2. *The control vector that minimizes* J_2 *is given by*

$$\mu_\star(t) = -R^{-1} h(x(t))^\top \rho(t) + \mu(x(t)). \tag{13}$$

Proof. Taking the derivative of (12) with respect to $\mu_\star$ gives

$$\begin{aligned} \frac{\partial}{\partial \mu_\star} J_2 &= \int_{t_i}^{t_i+T} \frac{\partial}{\partial \mu_\star} \left(\rho(t)^\top (f_2 - f_1) \right) + R(\mu_\star(t) - \mu(x(t))) dt \\ &= \int_{t_i}^{t_i+T} h(x(t))^\top \rho(t) + R(\mu_\star(t) - \mu(x(t))) dt. \end{aligned} \tag{14}$$

Since J_2 is convex in $\mu_\star$, we set the expression in (14) to zero and solve for $\mu_\star$ which gives us

$$\mu_\star(t) = -R^{-1} h(x(t))^\top \rho(t) + \mu(x(t))$$

which is a schedule of control values that reduce the objective for time $t \in [t_i, t_i + T]$. □

This controller reduces (5) for $\lambda > 0$ that is sufficiently small. The reduction in (5), ΔJ, by applying $\mu_\star(\tau)$ can be approximated as $\Delta J \approx \frac{\partial}{\partial \lambda} J \lambda \mid_{t=\tau}$. Ensuring that $\frac{\partial}{\partial \lambda} J < 0$ is an indicator that the robot is always actively pursuing data and reducing the objective (5).

Corollary 1. *Let us assume that* $\frac{\partial}{\partial \mu} \mathcal{H} \neq 0 \ \forall t \in [t_i, t_i + T]$*, where* $\mathcal{H}$ *is the control Hamiltonian. Then* $\frac{\partial}{\partial \lambda} J < 0 \ \forall \mu_\star(t) \in \mathcal{U}$ *where* $\mathcal{U}$ *is the control space.*

Proof. Inserting (13) into (6) gives

$$\begin{aligned} \frac{\partial}{\partial \lambda} J &= \rho(t)^\top (f_2 - f_1) \\ &= \rho(t)^\top \left(g(x(t)) + h(x(t)) \mu_\star(t) - g(x(t)) - h(x(t)) \mu(x(t)) \right). \end{aligned} \tag{15}$$

Because of the manner in which we chose to solve for $\mu_\star(t)$, $g(x)$ and $\mu(x(t))$ cancel out in (15). In addition, $\frac{\partial}{\partial \mu} \mathcal{H} \neq 0$ implies that $\frac{\partial}{\partial \lambda} J \neq 0$ and the policy $\mu(x(t))$ is not an optimizer of (5). As a result, we can further analyze $\frac{\partial}{\partial \lambda} J$ without the need to consider the policy $\mu(x)$. This gives us the following expression

$$\frac{\partial}{\partial \lambda} J = -\rho(t)^\top h(x(t)) R^{-1} h(x(t))^\top \rho(t)$$

which we can rewrite as

$$\frac{\partial}{\partial \lambda} J = -\|h(x(t))^\top \rho\|^2_{R^{-1}} < 0. \tag{16}$$

Thus, (16) shows us that $\frac{\partial}{\partial \lambda} J$ is always negative subject to the schedule of control vectors (13) and the objective is being reduced when $\mu_\star(t)$ is applied. □

We automate the switching between $\mu(x(t))$ and $\mu_\star(t)$ by choosing a τ and λ such that $\frac{\partial}{\partial \lambda} J$ is most negative and $\Delta J < 0$. This is done through the combination of choosing τ with a 1-dimensional optimization and solving for λ using a line search until $\Delta J < 0$ [16, 17]. By choosing $\lambda < T$ we can place a bound on how much our algorithm excites the dynamical system through Lyapunov analysis (Theorem 1).

Theorem 1. *Assume there exists a Lyapunov function $V(x)$ for (1) such that under the policy $\mu(x)$, $x(t)$ is asymptotically stable. That is, $\dot{V}(x) < 0 \; \forall \mu(x), x \in \mathcal{B}$ where $\mathcal{B} = \{x \in \mathbb{R}^n | \|x\| < r\}$ for $r > 0$. Then, given the schedule of control vectors (13) $\mu_\star(t)$ $\forall t \in [\tau, \tau+\lambda]$, $V(x(t)) - V(x(t), \mu(x(t))) \le \lambda\beta$, where $V(x(t), \mu(x(t)))$ is the Lyapunov function subject to the policy $\mu(x)$, and $\beta = \sup_{t\in[\tau,\tau+\lambda]} -\frac{\partial V}{\partial x} h(x(t)) R^{-1} h(x(t))^\top \rho(t)$.*

Proof. Writing the integral form of the Lyapunov function switching between $\mu(x(t))$ and $\mu_\star(t)$ at time τ for a duration of time λ starting at $x(0)$ gives

$$\begin{aligned} V(x(t)) = V(x(0)) \quad & + \int_0^t \dot{V}(x(s), \mu(x(s)))ds \\ = V(x(0)) \quad & + \int_0^\tau \dot{V}(x(s), \mu(x(s)))ds \\ & + \int_\tau^{\tau+\lambda} \dot{V}(x(s), \mu_\star(s))ds + \int_{\tau+\lambda}^t \dot{V}(x(s), \mu(x(s)))ds, \end{aligned} \tag{17}$$

where we explicitly write the dependency on $\mu(x(t))$ in $\dot{V}$. Using chain rule, we can write

$$\dot{V}(x,u) = \frac{\partial V}{\partial x} f(x,u) = \frac{\partial V}{\partial x} g(x) + \frac{\partial V}{\partial x} h(x) u. \tag{18}$$

By inserting (13) in (18) we can show the following identity:

$$\begin{aligned} \dot{V}(x, \mu_\star) &= \frac{\partial V}{\partial x} g(x) + \frac{\partial V}{\partial x} h(x) \mu_\star \\ &= \frac{\partial V}{\partial x} g(x) + \frac{\partial V}{\partial x} h(x) \left(-R^{-1} h(x)^\top \rho + \mu(x)\right) \\ &= \dot{V}(x, \mu(x)) - \frac{\partial V}{\partial x} h(x) R^{-1} h(x)^\top \rho. \end{aligned} \tag{19}$$

Using (19) in (17), we can show that

$$\begin{aligned} V(x(t)) &= V(x(0)) + \int_0^t \dot{V}(x(s), \mu(x(s)))ds - \int_\tau^{\tau+\lambda} \frac{\partial V}{\partial x} h(x(s)) R^{-1} h(x(s))^\top \rho(s) ds \\ &= V(x(t), \mu(x(t))) - \int_\tau^{\tau+\lambda} \frac{\partial V}{\partial x} h(x(s)) R^{-1} h(x(s))^\top \rho(s) ds \end{aligned} \tag{20}$$

where $V(x(t),\mu(x(t))) = V(x(0)) + \int_0^t \dot{V}(x(s),\mu(x(s)))ds$.

Letting the largest value of $\frac{\partial V}{\partial x} h(x(s))R^{-1}h(x(s))^\top \rho(s)$ be given by $\beta = \sup_{t\in[\tau,\tau+\lambda]} -\frac{\partial V}{\partial x} h(x(t))R^{-1}h(x(t))^\top \rho(t) > 0$, we can approximate (20) as

$$V(x(t)) = V(x(t),\mu(x(t)) - \int_\tau^{\tau+\lambda} \frac{\partial V}{\partial x} h(x(s))R^{-1}h(x(s))^\top \rho(s)ds \tag{21}$$

$$\leq V(x(t),\mu(x(t))) + \beta\lambda. \tag{22}$$

Subtracting both side by $V(x(t),\mu(x(t)))$ gives the upper bound on instability

$$V(x(t)) - V(x(t),\mu(x(t))) \leq \beta\lambda \tag{23}$$

for the active data collection process. □

By fixing the maximum value of λ, we can provide an upper bound to the change of the Lyapunov function during active data acquisition. Moreover, we can tune our control vector $\mu_\star(t)$ using the regularization value R such that as $\|R\| \to \infty$, $\beta \to 0$ and $\mu_\star(t) \to \mu(x(t))$. With this bound, we can guarantee Lyapunov attractiveness [18], where the system (1) is not Lyapunov stable, but rather there exists a time t such that the system (1) is guaranteed to return to a region of attraction where the system can be guided towards a stable equilibrium state x_0. This property will play an important role in examples in Section 5.

Definition 2. *A dynamical system (1) is Lyapunov attractive if at some time t, the trajectory of the system $x(t) \in \mathcal{C}(t) \subset \mathcal{B}$ where $\mathcal{C}(t) = \{x(t) \in \mathbb{R}^n | V(x) \leq c, \dot{V}(x(t)) < 0\}$ and $\lim_{t\to\infty} x(t) \to x_0$ such that x_0 is an equilibrium state.*

Theorem 2. *Given the schedule of control vectors (13) $\mu_\star(t)$ $\forall t \in [\tau, \tau+\lambda]$, the robotic system governed by the dynamics in (1) is Lyapunov attractive such that $\lim_{t\to\infty} x(t,\tau,\lambda) \to x_0$, where*

$$x(t,\tau,\lambda) = x(0) + \int_0^\tau f(x(s),\mu(x(s))ds + \int_\tau^{\tau+\lambda} f(x(s),\mu_\star(s)ds + \int_{\tau+\lambda}^t f(x(s),\mu(x(s))ds,$$

is the solution to switching between stable and exploratory motions for duration λ starting at time τ.

Proof. Assume there exists a Lyapunov function such that $\dot{V}(x) < 0$ under the policy $\mu(x)$. Moreover, assume that subject to the control vector $\mu_\star(t)$, the trajectory $x(\tau+\lambda) \in \mathcal{C}(\tau+\lambda) \subset \mathcal{B}$ where $\mathcal{C}(t) = \{x(t) \in \mathbb{R}^n | V(x) \leq c, \dot{V}(x(t),\mu(x(t)) < 0\}$ where $c > 0$. Using Theorem 1, the integral form of the Lyapunov function (17), and the identity (19), we can write

$$V(x(t)) = V(x(0)) + \int_0^t \dot{V}(x(s),\mu(x(s)))ds$$

$$- \int_\tau^{\tau+\lambda} \frac{\partial V}{\partial x} h(x(s))R^{-1}h(x(s))^\top \rho(s)ds \leq V(x(0)) - \gamma t + \beta\lambda, \tag{24}$$

where $-\gamma = \sup_{s\in[0,t]} \dot{V}(x(s), \mu(x(s))) < 0$. Since λ is fixed and β can be tuned by the matrix weight R, we can choose a t such that $\gamma t \gg \beta\lambda$. Thus, $\lim_{t\to\infty} V(x(t)) \to V(x_0)$ and $\lim_{t\to\infty} x(t, \tau, \lambda) \to x_0$, implying Lyapunov attractiveness, where $V(x_0)$ is the minimum of the Lyapunov function at the equilibrium state x_0. □

Asymptotic attractiveness shows us that the robot will return to a region where $V(x)$ will return to a minimum under policy $\mu(x)$, allowing the robot to actively explore and collect data safely. Moreover, we can choose the value of λ and $\mu_\star$ in automating the active data acquisition such that attractiveness always holds, giving us an algorithm that is safe for active data collection.

All that is left is to define a spatial distribution that actively selects which measurements are more informative to the learning task.

Measure of Data Importance for Model Learning Our goal is to provide a method that is general to any form of learning that requires a robot to actively seek out measurements through action. This may include area mapping or learning the dynamics of the robot. Thus, we use measures that allow the robot to quantify where in the search domain there exists useful data that needs to be collected. While there exists many measures that can select important data subject to a learning task, we use a measure of linear independence [7, 19, 20]. This measure is often used in sparse Gaussian processes [7, 20] where a data set $\mathcal{D} = \{x_i, y_i\}_{i=1}^{M}$ is comprised of M input measurements $x_i \in \mathbb{R}^v$ and M output measurements $y_i \in \mathbb{R}^c$ such that each data point maximizes the measure of linear independence. We use this measure of independence, also known as a measure of importance, to create a distribution for which the robot will provide area coverage in the search domain for active data collection.

As illustrated in [7], this is done by evaluating a new measurement x_{M+1}, y_{M+1} against the existing data points in $\mathcal{D}$ given the structure of the model that is being learned.

Definition 3. *The importance measure $\delta \in \mathbb{R}^+$ for a new measurement pair $\{x_{M+1}, y_{M+1}\}$ is given by*

$$\delta = k(x_{m+1}, x_{m+1}) - \mathbf{k}^\top \mathbf{a} \tag{25}$$

which is the solution to $\delta = \| \sum_{i=1}^{M} a_i \phi(x_i) - \phi(x_{M+1}) \|^2$, where $\phi(x)$ are the basis functions (also known as feature vectors) [1], a_i is the coefficient of linear dependence, the matrix $K \in \mathbb{R}^{M\times M}$ is known as the kernel matrix with elements $K_{i,j} = k(x_i, x_j)$ such that $k : \mathbb{R}^{v\times v} \to \mathbb{R}$ is the kernel function given by the inner product $k(x_i, x_j) = \langle \phi(x_i), \phi(x_j) \rangle$, $\mathbf{k} = [k(x_1, x_{m+1}), k(x_2, x_{m+1}), \ldots, k(x_m, x_{m+1})]^\top$, and $\mathbf{a} = K^{-1}\mathbf{k}$.

[1] This feature vector can be anything from a Fourier set of basis functions or a neural network. In addition, we can parameterize the functions $\phi(x) = \phi(x, \theta)$ and have the functions change over time.

The value δ provides a measure of how well the point x_{M+1} can be represented given the existing data set and structure of the model being learned. Note that this measure will be computationally intractable for very large M. Instead, other measures like the expected information density derived from the Fisher information matrix [12, 21] can be used if the learning task has a model that is parameterized by a set of parameters θ. Since $\delta > 0$, we define an importance distribution for which the robot will use generate area coverage.

Definition 4. *The importance distribution is*

$$p(s) = \frac{1}{\eta}\left(k(s,s) - \mathbf{k}(s)^\top \mathbf{a}(s)\right) \tag{26}$$

where $\eta = \int_{\mathcal{X}^v} k(s,s) - \mathbf{k}(s)^\top \mathbf{a}(s) ds$*, and* $\mathbf{k}$, $\mathbf{a}$ *are functions of points* $s \in \mathcal{X}^v$*.*

Note that $p(s)$ will change as $\mathcal{D}$ is augmented or pruned. If at any time $\delta > \delta_i$ for $i = [1, \ldots, M]$, we remove the i^{th} point with the lowest δ value and add in the new data point.

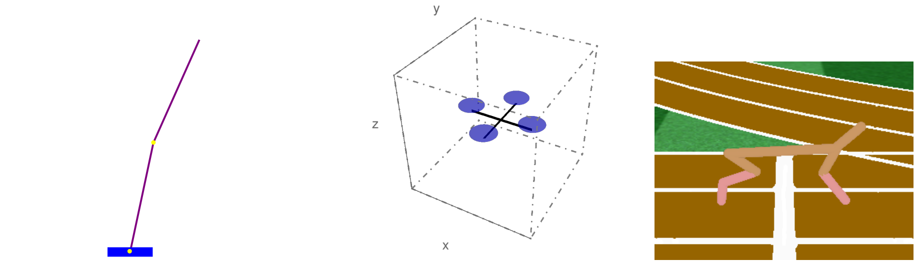
We provide an outline of our method in Algorithm 1 for online data acquisition. The following section evaluates our method on various simulated environments.

Algorithm 1 Active Data Acquisition from Equilibrium

```
initialize: local dynamics model, initial condition $x(t_0)$, initial equilibrium policy $\mu(x)$, learning task model structure $\phi(x)$.
for $i = 0, \ldots, \infty$ do
    simulate $x(t)$ with $\mu(x(t))$ from $x(t_i)$ using dynamics model $f(x,u)$
    calculate $\rho(t)$ and $\frac{\partial}{\partial \lambda} J$
    compute control $\mu_\star(t)$ for $t \in [t_i, t_i + T]$
    choose $\tau, \lambda$ that minimizes $\frac{\partial}{\partial \lambda} J$
    apply $\mu_\star(\tau)$ if $t \in [\tau, \tau + \lambda]$ else apply $\mu(x(t))$ to robot
    sample state $x(t_{i+1})$ and measurement $y(t_{i+1})$
    verify importance $\delta$ and update $p(s)$ if $\delta > \delta_i \forall i \in [1, \ldots, M]$
end for
```

5 Simulated Examples

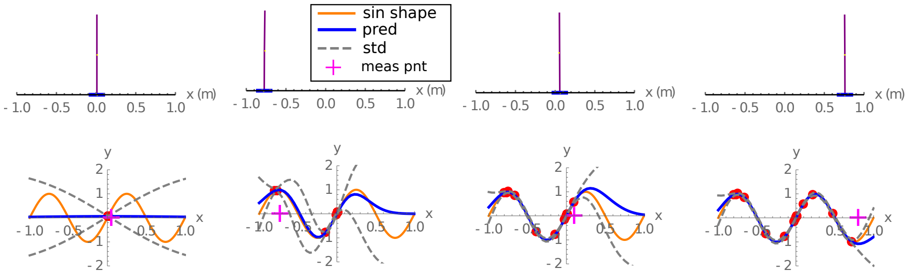
In this section, we illustrate examples of Algorithm 1 for different examples that may be encountered in robotics. Figure 1 depicts three robotic systems on which we base our examples. In the first example, we use a cart double pendulum for use in area coverage for shape estimation. In the second and third example, we use a 22 dimensional quadrotor [22] and a 26 dimensional half-cheetah model from Roboschool [23] for learning a dynamics model of the robotic systems by exploring in the state-space. For implementation details, including parameters used, we refer the reader to the appendix.



(a) Cart Double Pendulum (b) Quadrotor (c) Half-Cheetah

Fig. 1: Simulated experimental environments (a) cart double pendulum and (b) quadcopter, and (c) half-cheetah. The results for each system may be seen in the accompanying multimedia supplement.

Shape Estimation while Stabilizing Cart Double Pendulum Our first example demonstrates the functionality of our algorithm for estimating a sinusoidal shape while simultaneously balancing a cart double pendulum in its upright position. The purpose of this example is to show that our method can synthesize actions that ensures the cart double pendulum is maintained upright while actively collecting data for estimating the shape. This example also serves the purpose of illustrating that our method can safely automate choosing when to stabilize and when to explore for data using approximate linear models of the robot dynamics and stabilizing policies derived from the approximate models.



(a) $t = 0$ (b) $t = 6$ (c) $t = 14$ (d) $t = 20$

Fig. 2: Time series snap-shots of cart double pendulum actively sampling and estimating the shape underneath. The uncertainty (dashed gray lines) calculated from the collected data set drives the exploratory motion of the cart double pendulum while our method ensures that the cart double pendulum is maintained in its upright equilibrium state.

The measurements of the height of the sinusoidal shape are collected through x position of the cart (illustrated in Fig. 2 as the magenta crosshair underneath the cart). A Gaussian process with an radial basis function (RBF) kernel [7] is then used to estimate the function and provide the distribution used for exploration. The underlying importance distribution (26) is updated as the data set is pruned to include new informative measurements.

As a result of Algorithm 1, the robot will spend time where there is a high probability of acquiring informative data. This results is the shape reconstruction shown in Fig. 2 using a limited fixed set of data ($M = 50$).

We analyze our algorithm by observing a candidate Lyapunov function (energy). Figure 3 depicts the value of the Lyapunov function over the time window of the cart double pendulum collecting data for estimating shape. The control vector $\mu_\star(t)$ over the application time $t \in [\tau, \tau + \lambda]$ increases the overall energy in the system (due to exploration). Since we include a regularization term R that ensures $\mu_\star$ does not deviate too far from the equilibrium policy $\mu(x)$, the cart double pendulum is able to stabilize itself, eventually returning to an equilibrium state and ensuring stability, illustrating the Lyapunov attractiveness property proven in Theorem 2.

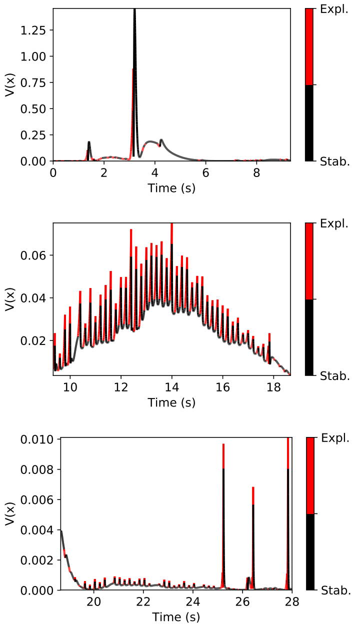


Fig. 3: Lyapunov function for the cart double pendulum with upright equilibrium. The red line indicates when the active exploration control is applied. Lyapunov attractiveness property is illustrated through automatic switching of the exploration process.

Learning Dynamics of Quadrotor Our next example illustrates active data acquisition in the state-space of a 22 degree of freedom quadrotor vehicle shown in Fig. 1a. The results are averaged across 30 trials with random initial conditions sampled uniformly in the body angular and linear velocities $\omega, v \sim \mathcal{U}[-0.01, 0.01]$ where $\mathcal{U}$ is a uniform distribution.

The goal for this quadrotor is to maintain hovering height while collecting data in order to learn the dynamics model $f(x, u)$. In this example, a linear approximation of the dynamics centered at the hovering height is used as the local dynamics approximation on which Algorithm 1 is based. We then generate a LQR controller with the approximate dynamics which we use as the equilibrium policy, The input data we collect is the state $x(t_i) = x_i$ and control $u(t_i) = u_i$ and the output data is $(x_{i+1} - x_i)/(t_{i+1} - t_i)$ which approximates the function $\frac{\Delta x}{\Delta t} \approx \dot{x} = f(x, u)$ [24]. An incremental sparse Gaussian process [7] with a radial basis function kernel is used to generate a learned model of the dynamics using a data set of $M = 80$ and to specify the importance measure (3).

Figure 4 (a) and Figure 4 (b) illustrates the modeling error and the minimum importance value within the data set using our method and the equilibrium policy with uniformly added added noise at 33% of the saturation limit. Our method sequences and automates the process of choosing when it is best to explore and to stabilize by taking into account the approximate dynamics and the equilibrium policy. As a result, a robot is capable of acquiring informative data that improves the prediction of the nonlinear dynamic model of the quadrotor. In

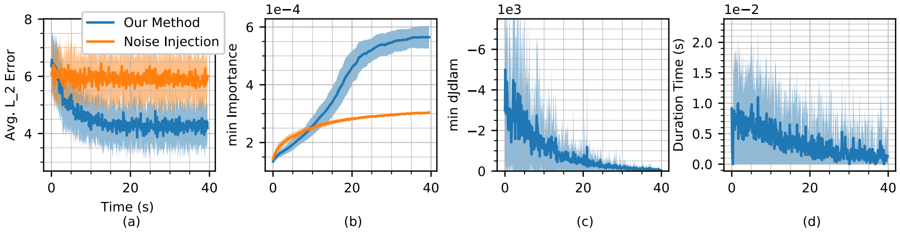


Fig. 4: (a) Average L_2 error of the learned dynamics model evaluated on 10 uniformly distributed random samples between $[-1, 1]$ in the state and action space. (b) Minimum importance measure from the collected data set using our method compared against injected noise. (c) Minimum value of the mode insertion gradient evaluated at the chosen $\mu_\star(\tau)$ value. (d) Calculated duration time of control $\mu_\star(\tau)$.

contrast, adding noise to the control input (often referred to as "motor babble" [24]) does not have temporal dependencies. That is, each new sample does not have information from the previous samples and cannot effectively explore the state-space.

As the robot continues to explore, the value of the mode insertion gradient (6) decreases as does the duration time λ as shown in Fig. 4 (c) and (d). This implies that the robot is sufficiently reducing the objective for area coverage and the equilibrium policy begins to take over to stabilize the robot. This is a result of taking into account the local stability of the robotic system while generating exploratory actions.

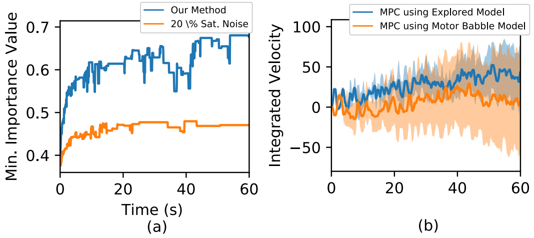


Fig. 5: (a) Comparison of the minimum importance measure of the the data set for the half-cheetah example from a stable standing policy with added 20% saturation limit noise and our approach for active data acquisition. (b) Integrated forward velocity values from using the learned half-cheetah dynamics model in a model-predictive control setting with standard deviation illustrated for 5 trials. Our method is shown to collect data which provides a better dynamic model and a positive net forward velocity with reduced variance.

Learning to Gallop In this last example, we consider applications of Algorithm 1 for systems with dynamic models and policies that are learned. We use the half-cheetah from the roboschool environment [23] for the task of learning a dynamics model in order to control the robot to gallop forward.

We first learn a simple standing upright policy using the augmented random search (ARS) method [25]. In that process, we collect the state and action data to compute a linear approximation using least-squares for the local dynamics. Then Algorithm 1 is applied using an incremental sparse Gaussian process using an RBF kernel to generate a dynamics model from data as well as provide the importance measure using a set of $M = 40$ data points. The input-output data structure maps input $(x(t_i), u(t_i))$ to the change in state $\frac{\Delta x}{\Delta t}$. Our running cost $\ell(x, u)$ is set to maintain the half-cheetah upright. After the Gaussian process model is learned, we use the generated model in the prediction of the forward dynamics as a replacement for the initial dynamics model.

As shown in Fig. 5, our method collects informative data while respecting the standing upright policy when compared to noisy inputs. We compared the two learned models using our controller with $D_{\text{KL}} = 0$ and the running cost $\ell(x, u)$ set to maximize the forward velocity of the half-cheetah. We show those results in Fig. 5 over 5 runs of our algorithm at different initial states. Our method provides a learned model that has overall positive integrated velocity (forward movement). While our method is more complex than simply adding noise, it provides stability guarantees based on known policies in order to explore and collect data.

6 Conclusion

Algorithm 1 enables robots to actively seek out informative data based on the learning task while maintaining stability using equilibrium policies. Our method generates area coverage using a KL-divergence measure in order to enable robots to actively seek out informative data. Moreover, by using a hybrid systems theory approach to generating area coverage, we were able to incorporate equilibrium policies in order to provide stability guarantees even with the model of the robot dynamics only locally known. Last, we provide examples that illustrate the benefits of our approach for active data acquisition for learning tasks.

Bibliography

[1] Petar Kormushev, Sylvain Calinon, and Darwin G Caldwell. Robot motor skill coordination with em-based reinforcement learning. In *International Conference on Intelligent Robots and Systems*, pages 3232–3237, 2010.

[2] René Felix Reinhart. Autonomous exploration of motor skills by skill babbling. *Autonomous Robots*, 41(7):1521–1537, 2017.

[3] Christopher D McKinnon and Angela P Schoellig. Learning multimodal models for robot dynamics online with a mixture of Gaussian process experts. In *International Conference on Robotics and Automation*, pages 322–328, 2017.

[4] Jie Tan, Tingnan Zhang, Erwin Coumans, Atil Iscen, Yunfei Bai, Danijar Hafner, Steven Bohez, and Vincent Vanhoucke. Sim-to-real: Learning agile locomotion for quadruped robots. In *Proceedings of Robotics: Science and Systems*, 2018. doi: 10.15607/RSS.2018.XIV.010.

[5] Alonso Marco, Felix Berkenkamp, Philipp Hennig, Angela P Schoellig, Andreas Krause, Stefan Schaal, and Sebastian Trimpe. Virtual vs. real: Trading off simulations and physical experiments in reinforcement learning with bayesian optimization. In *International Conference on Robotics and Automation*, pages 1557–1563, 2017.

[6] Mac Schwager, Philip Dames, Daniela Rus, and Vijay Kumar. A multi-robot control policy for information gathering in the presence of unknown hazards. In *Robotics research*, pages 455–472. 2017.

[7] Duy Nguyen-Tuong and Jan Peters. Incremental online sparsification for model learning in real-time robot control. *Neurocomputing*, 74(11):1859–1867, 2011.

[8] Dariusz Ucinski. *Optimal measurement methods for distributed parameter system identification.* CRC Press, 2004.

[9] Felix Berkenkamp, Matteo Turchetta, Angela Schoellig, and Andreas Krause. Safe model-based reinforcement learning with stability guarantees. In *Advances in Neural Information Processing Systems*, pages 908–918, 2017.

[10] Tsen-Chang Lin and Yen-Chen Liu. Direct learning coverage control based on expectation maximization in wireless sensor and robot network. In *Conference on Control Technology and Applications*, pages 1784–1790, 2017.

[11] Frederic Bourgault, Alexei A Makarenko, Stefan B Williams, Ben Grocholsky, and Hugh F Durrant-Whyte. Information based adaptive robotic exploration. In *International Conference on Intelligent Robots and Systems*, volume 1, pages 540–545, 2002.

[12] Lauren M Miller, Yonatan Silverman, Malcolm A MacIver, and Todd D Murphey. Ergodic exploration of distributed information. *IEEE Transactions on Robotics*, 32(1):36–52, 2016.

[13] Elif Ayvali, Hadi Salman, and Howie Choset. Ergodic coverage in constrained environments using stochastic trajectory optimization. In *International Conference on Intelligent Robots and Systems*, pages 5204–5210, 2017.

[14] Randolf Arnold and Andreas Wellerding. On the sobolev distance of convex bodies. *aequationes mathematicae*, 44(1):72–83, 1992.

[15] Henrik Axelsson, Y Wardi, Magnus Egerstedt, and EI Verriest. Gradient descent approach to optimal mode scheduling in hybrid dynamical systems. *Journal of Optimization Theory and Applications*, 136(2):167–186, 2008.

[16] Anastasia Mavrommati, Emmanouil Tzorakoleftherakis, Ian Abraham, and Todd D Murphey. Real-time area coverage and target localization using receding-horizon ergodic exploration. *IEEE Transactions on Robotics*, 34(1):62–80, 2018.

[17] Ian Abraham and Todd D. Murphey. Decentralized ergodic control: Distribution-driven sensing and exploration for multiagent systems. *IEEE Robotics and Automation Letters*, 3(4):2987–2994, 2018.

[18] Andrey Polyakov and Leonid Fridman. Stability notions and lyapunov functions for sliding mode control systems. *Journal of the Franklin Institute*, 351(4):1831–1865, 2014.

[19] Bernhard Scholkopf, Sebastian Mika, Chris JC Burges, Philipp Knirsch, K-R Muller, Gunnar Ratsch, and Alex J Smola. Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, 10(5):1000–1017, 1999.

[20] Xinyan Yan, Vadim Indelman, and Byron Boots. Incremental sparse gp regression for continuous-time trajectory estimation and mapping. *Robotics and Autonomous Systems*, 87:120–132, 2017.

[21] AF Emery and Aleksey V Nenarokomov. Optimal experiment design. *Measurement Science and Technology*, 9(6):864, 1998.

[22] Taosha Fan and Todd Murphey. Online feedback control for input-saturated robotic systems on Lie groups. In *Proceedings of Robotics: Science and Systems*, June 2016. doi: 10.15607/RSS.2016.XII.027.

[23] Oleg Klimov and John Shulman. Roboschool. `https://github.com/openai/roboschool`, 2017.

[24] Anusha Nagabandi, Gregory Kahn, Ronald S Fearing, and Sergey Levine. Neural network dynamics for model-based deep reinforcement learning with model-free fine-tuning. *arXiv preprint arXiv:1708.02596*, 2017.

[25] Horia Mania, Aurelia Guy, and Benjamin Recht. Simple random search provides a competitive approach to reinforcement learning. *arXiv preprint arXiv:1803.07055*, 2018.